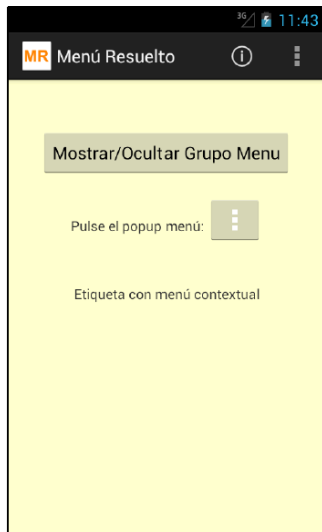


Solución al ejercicio propuesto menús. Como se indica en el documento la aplicación constará de una única activity que tendrá un button, un imagen button y dos textView. El layout correspondiente a la ActivityMain sería:



```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="vertical"
    android:paddingLeft="16dp"
    android:paddingRight="16dp"
    android:paddingTop="100dp"
    android:paddingBottom="16dp"
    tools:context=".MyActivity"
    android:gravity="center"
    android:weightSum="1"
    android:id="@+id/layout1">

    <Button
        android:layout_width="319dp"
        android:layout_height="wrap_content"
        android:text="Mostrar/Ocultar Grupo Menú"
        android:id="@+id/button"
        android:layout_alignBaseline="@+id/textView"
        android:layout_alignBottom="@+id/textView"
        android:layout_alignParentLeft="true"
        android:layout_alignParentStart="true" />

    <LinearLayout
        android:orientation="horizontal"
        android:layout_width="fill_parent"
        android:layout_height="168dp"
        android:gravity="center"
        android:layout_weight="4.03"
        android:id="@+id/layout2">

        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Pulse el popup menú"
            android:id="@+id/textView2"
            android:textSize="20dp" />

        <ImageButton
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:id="@+id/imageButton"
            android:src="@drawable/popup" />

    </LinearLayout>

    <TextView
        android:text="Etiqueta menú contextual"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@+id/textView"
        android:layout_alignRight="@+id/button"
        android:layout_alignEnd="@+id/button"
        android:textSize="30dp"
        />

</LinearLayout>
```

Los XML de los menús podrían ser los siguientes. El referente al TextView con las opciones de **Editar** y **Restaurar** el ítem, podría ser como el siguiente:

```
<?xml version="1.0" encoding="utf-8"?>

<menu xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/menuPopup">
    <item android:id="@+id/editar"
        android:title="Editar"/>

    <item android:id="@+id/restaurar"
        android:title="Restaurar"/>
</menu>
```

Mientras que el menú para el popput podría ser:

```
<?xml version="1.0" encoding="utf-8" ?>

<menu xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/menuPopup">
    <item android:id="@+id/cambiarTexto"
        android:title="Cambiar texto"/>

    <item android:id="@+id/cambiarColor"
        android:title="Cambiar color"/>
</menu>
```

El menú de opciones de la main, con el grupo y demás podría ser de la siguiente manera:

```
<menu xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    tools:context=".MyActivity">
    <item android:id="@+id/informacion"
        android:icon="@drawable/info"
        android:title="@string/action_settings"
        android:orderInCategory="100"
        app:showAsAction="always" />
    <group android:id="@+id/grupo">
        <item android:id="@+id/t12"
            android:title="Tamaño 12"
            android:orderInCategory="100"
            app:showAsAction="never" />
        <item android:id="@+id/t18"
            android:title="Tamaño 18"
            android:orderInCategory="100"
            app:showAsAction="never" />
        <item android:id="@+id/t24"
            android:title="Tamaño 24"
            android:orderInCategory="100"
            app:showAsAction="never" />
    </group>
    <item android:id="@+id/estilo"
        android:title=" Estilo..."
        android:orderInCategory="100"
        app:showAsAction="never" >
        <menu android:id="@+id/estiloSubMenu">
            <item android:id="@+id/negrita"
                android:title=" Negrita"
                android:icon="@drawable/item"
                android:orderInCategory="100"
                app:showAsAction="never"/>
            <item android:id="@+id/cursiva"
                android:title="Cursiva"
                android:icon="@drawable/item"
                android:orderInCategory="100"
                app:showAsAction="never"/>
        </menu>
    </item>
</menu>
```

En cuanto al código de la main, tendremos en el onCreate la declaración de variables y las escuchas del menú contextual y los listeners de los diferentes botones y textos sobre los que queremos mostrar la acción del menú.

```
public class MyActivity extends Activity implements View.OnClickListener{
    Button botonMostrar;
    ImageButton botonPopup;
    Boolean visible;
    TextView texto;
    PopupMenu menuPopup;
    MenuItem itemRestaurar;
    String cadena;
    @Override
}
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_my);
    texto=(TextView) findViewById(R.id.textview);

    itemRestaurar=(MenuItem) findViewById(R.id.informacion);
    cadena=texto.getText().toString();
    visible=true;
    botonMostrar=(Button) findViewById(R.id.button);
    botonPopup=(ImageButton) findViewById(R.id.imagebutton);
    botonPopup.setOnClickListener(this);
    botonMostrar.setOnClickListener(this);
    registerContextMenu(texto);
}
```

```

public void onClick(View view) {
    switch (view.getId())
    {
        case R.id.button:
            if(visible) visible=false;
            else visible=true;
            break;
        case R.id.imageButton:
            menuPopup=new PopupMenu(getApplicationContext(),view);
            menuPopup.getMenuInflater().inflate(R.menu.menu_popup,menuPopup.getMenu());
            menuPopup.setOnMenuItemClickListener(new MenuItemClickListener() {
                switch (menuItem.getItemId())
                {
                    case R.id.cambiarColor:
                        texto.setTextColor(Color.rgb(55,00,33));
                        break;
                    case R.id.cambiarTexto:
                        texto.setText("Has cambiado el texto");
                        break;
                }
            });
            return false;
        }
    }
    menuPopup.show();
}
}

```

Menú popup

Como podemos ver en la imagen superior, en el caso de pulsar sobre el ImageButton, se lanzará un menú popup con las opciones de cambiar color y texto.

Menú de opciones

Para el menú de opciones inflaremos el menú correspondiente en el método onCreateOptionsMenu. Como las opciones 2 a 4 pertenecen a un mismo grupo que se puede mostrar u ocultar pulsando sobre el botón “Mostrar/Ocultar Grupo Menu”, implementaremos un boolean, imagen superior, que se intercambiará entre true y false en cada pulsación. En el método onPrepareOptionsMenu (método que se llama justo antes de visualizar el menú), asignaremos la visibilidad correspondiente. En el onOptionsItemSelected se realizarán las acciones correspondientes dependiendo del elemento pulsado.

```

@Override
public boolean onCreateOptionsMenu(Menu menu) {
    // Inflate the menu; this adds items to the action bar if it is present.
    getMenuInflater().inflate(R.menu.my, menu);
    return true;
}

@Override
public boolean onPrepareOptionsMenu(Menu menu) {
    menu.setGroupVisible(R.id.grupo, visible);
    return super.onPrepareOptionsMenu(menu);
}

@Override
public boolean onOptionsItemSelected(MenuItem item) {
    super.onOptionsItemSelected(item);
    int id = item.getItemId();
    if(id==R.id.informacion) Toast.makeText(this,"Pulsa prolongadamente sobre la etiqueta " +
        "para restaurar", Toast.LENGTH_LONG).show();
    else if(id==R.id.cursiva) texto.setTypeface(null, Typeface.ITALIC);
    else if(id==R.id.negrita) texto.setTypeface(null, Typeface.BOLD);
    else if(id==R.id.t12) texto.setTextSize(12);
    else if(id==R.id.t18) texto.setTextSize(18);
    else if(id==R.id.t24) texto.setTextSize(24);
    else return false;
    return true;
}
}

```

Menú contextual

En el caso del menú contextual, asignaremos el escuchador sobre el elemento correspondiente y después pasaremos a implementar los métodos correspondientes:

```
registerForContextMenu(texto);
```

```
@Override
public void onCreateContextMenu(ContextMenu menu, View v, ContextMenu.ContextMenuInfo menuInfo) {
    super.onCreateContextMenu(menu, v, menuInfo);
    MenuInflater inflater=getMenuInflater();
    inflater.inflate(R.menu.menu_context,menu);
}

@Override
public boolean onContextItemSelected(MenuItem item) {
    switch (item.getItemId())
    {
        case R.id.editar:
            Toast.makeText(MyActivity.this,"Has pulsado editar",Toast.LENGTH_LONG).show();
            break;
        case R.id.restaurar:
            texto.setText(cadena);
            texto.setTextSize(20);
            texto.setTextColor(Color.BLACK);
            texto.setTypeface(null,Typeface.NORMAL);

            break;
    }
    return super.onContextItemSelected(item);
}
```